WHAT IS CLAIMED IS:

1. An information processing method whereby one or more tasks are executed under an executing environment provided by an operating system, said method comprising:

a step for executing mutual verification between said operating system and a task at the time of activating said task.

2. An information processing method according to Claim 1, wherein said mutual verification step is performed using a key given by a user describing a task.

3. An information processing method whereby one or more tasks are executed under an executing environment provided by an operating system, said method comprising:

a step for executing mutual verification between said operating system and a task at the time of said task requesting service of said operating system.

4. An information processing method according to Claim 3, wherein said mutual verification step is performed by using data created by enciphering predetermined data with key information used for a first mutual verification as an execution key for a second mutual verification, and wherein

said second mutual verification is performed following said first mutual verification.

5. An information processing method whereby one or more tasks are executed under an executing environment provided by an operating system, said method comprising:

a step for said operating system to verify a task based on a mutual verification key held by said task;

a step for said operating system to encipher the stack pointer of said task with said mutual verification key, and return it to said task; and

a step for said task to decipher said stack pointer enciphered with said mutual verification key and perform verification.

6. An information processing method according to Claim 5, wherein, in the event that said verification is successful, said operating system and task hold the stack pointer enciphered with said mutual verification key as an executing key to be used for subsequent verification steps.

7. An information processing method whereby one or more tasks are executed under an executing environment provided by an operating system, said method comprising:

a step for a task to request service of said operating

system, with an executing key attached;

a step for said operating system to verify said task based on said executing key; and

a step for said operating system, in response to success of said verification, to execute services commissioned, and to encipher the stack pointer of said task with said executing key to generate a next executing key.

8. A program for a computer to execute information processing whereby one or more tasks are executed under an executing environment provided by an operating system, said processing comprising:

a step for said operating system to verify a task based on a mutual verification key held by said task;

a step for said operating system to encipher the stack pointer of said task with mutual verification key, and return it to said task; and

a step for said task to compound and verify the stack pointer enciphered with said mutual verification key;

wherein, in the event that said verification procedures are successful, said operating system and task hold the stack pointer enciphered with said mutual verification key as an executing key to be used for subsequent verification steps.

9. A program for a computer to execute information processing whereby one or more tasks are executed under an executing environment provided by an operating system, said processing comprising:

a step for a task to request service of said operating system, with a first executing key attached;

a step for said operating system to verify said task based on said first executing key; and

a step for said operating system, in response to success of said verification, to execute services requested, and to encipher the stack pointer of said task with said executing key to generate a second executing key to be used next time.

10. A communication method between tasks executed on an operating system, said method comprising:

a step for managing the security level of tasks themselves and mutual verification keys for mutual verification between tasks and said operating system, in a table format at said operating system side, wherein said security level has a first level which is secure and a second level which is not secure;

a step for reading and writing blocks of tasks of said first level and blocks of tasks of said second level separately into a secure memory block and a non-secure

memory block, respectively;

a step for providing a first buffer on a first task of a task of said first level, and a second buffer on a second task of a task of said second level, and providing within said operating system memory area for storing data and memory area for storing management information;

a step for specifying at said first task an ID and an address appropriated at said first task side, judging at said operating system side which memory block to use based on the security level of said first task and the security level of a first function, and, in the event that the security level of said first task and the level for executing said first task are secure, management information is written to said security memory block and data is written as enciphered contents with the ID, address value of management information, and address value of the data body, as a key; and

a step for specifying at said second task an ID the same as said second task and an address appropriated at said second task side, judging at said operating system side which secure memory block to use based on the security level of said second task and the security level of a second function, and, in the event that the security level of said second task and the level for executing said second level are secure, data addressed to said second task managed in

said secure memory block is searched, and the contents of the buffer where said data exists is copied onto said second task having been deciphered with the ID, address value of management information, and address value of the data body, as a key.

11. A communication method according to Claim 10, wherein said task is any of a semaphore, event flag, or mailbox.

12. A communication method according to Claim 10, wherein said task is a semaphore, and said first task is returning of resources and said second task is standing by to capture resources.

13. A communication method according to Claim 10, wherein said task is an event flag, and said first task is setting an event flag and said second task is clearing an event flag.

14. A communication method according to Claim 10, wherein said task is a mailbox, and said first task is transmitting data and said second task is receiving data.

15. A communication method according to Claim 10,

wherein said verification is performed by collating whether or not a key each task has is the same as the key managed at the operating system side.

16. A communication method according to Claim 10, wherein said method is carried out by memory managing means which performs reading and writing discriminatorily between said secure memory block and said non-secure memory block.

17. A communication method according to Claim 16, wherein said memory managing means comprise hardware capable of setting access permission for each block of said memory blocks according to security level.

18. A communication method according to Claim 16, wherein said memory managing means are not capable of reading from or writing to the memory block of a first level with regard to a first task or a second task in the second level.

19. A communication method according to Claim 16, wherein said operating system performs management of security levels for each task and management of said memory blocks via said memory managing means in a centralized manner.

20. A communication method according to Claim 14, wherein said management information comprises mail size and a mail body pointer.